

500,197

10/500197

JP 02/13442

#2

日本国特許庁
JAPAN PATENT OFFICE

24 JUN 2004

28.02.03

38

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2001年12月28日

出願番号

Application Number:

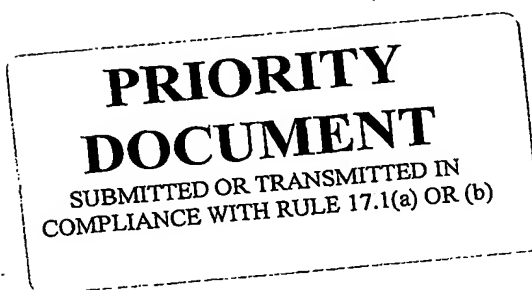
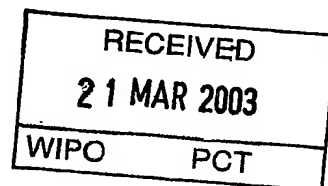
特願2001-401462

ST.10/C]:

[JP2001-401462]

出願人
Applicant(s):

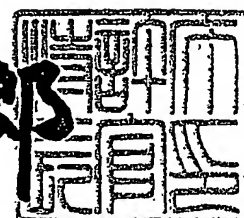
東京エレクトロニクス株式会社
西原 明法



2003年 1月17日

特許庁長官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2002-3107139

【書類名】 特許願

【整理番号】 JPP014008

【提出日】 平成13年12月28日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/00

【発明者】

【住所又は居所】 神奈川県横浜市都筑区東方町 1 番地 東京エレクトロニクス株式会社内

【氏名】 三田 高司

【特許出願人】

【識別番号】 500323188

【氏名又は名称】 東京エレクトロニクス株式会社

【特許出願人】

【識別番号】 501186977

【氏名又は名称】 西原 明法

【代理人】

【識別番号】 100095407

【弁理士】

【氏名又は名称】 木村 満

【手数料の表示】

【予納台帳番号】 038380

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0014440

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 演算システム

【特許請求の範囲】

【請求項 1】

ルックアップテーブルを含んだ複数のプログラムモジュールを各々記憶する複数のプログラム記憶手段と、

各前記プログラム記憶手段のいずれかを選択する選択手段と、

複数の論理回路を含み、前記選択手段に選択されたプログラム記憶手段に記憶されているプログラムモジュール中の命令に従った信号を前記複数の論理回路の 1 以上に入力することで、該プログラムモジュールに応じた演算を実行する論理演算手段と、

を備えることを特徴とする演算システム。

【請求項 2】

前記選択手段は、シフトレジスタより構成されている、

ことを特徴とする請求項 1 に記載の演算システム。

【請求項 3】

各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記憶するものであって、

前記選択手段は、所定のプログラム記憶手段が記憶するプログラムモジュールを選択するものであって、各前記プログラムモジュールを、各前記プログラム記憶手段相互間でサイクリックに転送させる、

ことを特徴とする請求項 1 又は 2 に記載の演算システム。

【請求項 4】

各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記憶するものであって、

前記プログラム記憶手段にプログラムモジュールをロードするロード手段を更に備える、

ことを特徴とする請求項 1、2 又は 3 に記載の演算システム。

【請求項 5】

前記論理演算手段の内部状態を退避する退避手段と、

所定の条件が成立した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、他のプログラムモジュールを前記ロード手段にロードさせ、該他のプログラムモジュールを記憶したプログラム記憶手段を前記選択手段に選択させ、該他のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、元のプログラムモジュールに応じた演算の実行に復帰させる制御手段と、

を更に備えることを特徴とする請求項 4 に記載の演算システム。

【請求項 6】

前記複数のプログラムモジュールのうちの少なくとも一部のプログラムモジュールは、他のプログラムモジュールを呼び出す機能を含み、

前記論理演算手段で演算を実行しているプログラムモジュール中の命令における他のプログラムモジュールの呼び出しを検出する呼び出し検出手段をさらに備え、

前記制御手段は、前記呼び出し検出手段が他のプログラムモジュールの呼び出しを検出した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、呼び出し先のプログラムモジュールが前記プログラム記憶手段に記憶されていないときは該呼び出し先のプログラムモジュールを前記ロード手段にロードさせ、該呼び出し先のプログラムモジュールを記憶しているプログラム記憶手段を前記選択手段に選択させ、該呼び出し先のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、呼び出し元のプログラムモジュールに応じた演算の実行に復帰させることを特徴とする請求項 5 に記載の演算システム。

【請求項 7】

前記制御手段によって実行が切り替えられるプログラムモジュール間において引数を受け渡すための引数受け渡し手段をさらに備える

ことを特徴とする請求項 5 又は 6 に記載の演算システム。

【請求項 8】

前記退避手段は、先入れ後出し方式のスタックによって構成される

ことを特徴とする請求項 5、6 又は 7 に記載の演算システム。

【請求項 9】

前記プログラム記憶手段に記憶された各プログラムモジュールは、前記論理演算手段を構成する論理回路に入力する信号に応じたコードによって構成されている

ことを特徴とする請求項 1 乃至 8 のいずれか 1 項に記載の演算システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、プログラムの実行をハードウェアで直接的に実現できる演算システムに関し、特に大規模プログラムの実行に適した演算システムに関する。

【0002】

【従来の技術】

現在の汎用コンピュータは、CPU (Central Processing Unit) がメモリに記憶されたプログラム中の命令を順次解釈しながら、演算を進めていく。CPU は、プログラムで実行すべき演算をソフトウェアで実現するものであり、必ずしもその演算に対して最適なハードウェア構成となっていないため、最終的な演算結果を得るまでに多くのオーバーヘッドが存在する。

【0003】

これに対して、プログラムの実行をハードウェアで直接的に実現するための技術として、例えば、特表平 8-504285 号公報 (国際公開 WO 94/10627 号公報) や特表 2000-516418 号公報 (国際公開 WO 98/08306 号公報) に示されているような、フィールドプログラマブルゲートアレイ (FPGA) を利用した演算システムが知られている。

【0004】

FPGA は、プログラムとして論理データを与えることで論理回路間の結線論理を変更し、これによってハードウェア的に演算結果を得ることをできるようにしたものである。FPGA を利用して演算を行うことによって、特定の演算専用構成されたハードウェア回路ほどは高速ではないが、従来の汎用コンピュータ

のようなCPUによる演算に比べると、非常に高速に演算結果を得ることができる。

【0005】

【発明が解決しようとする課題】

ところで、現在の汎用コンピュータで実行されているプログラム、特に大規模なプログラムは、複数のモジュールに分割して作成されている。そして、1のプログラムモジュールが他のプログラムモジュールを呼び出しながら、全体としてのプログラムの実行を進めていくようになっている。こうしてプログラムモジュール別に開発を進めたり、各プログラムモジュールを部品として利用したりすることにより、プログラムの開発期間を短縮することができる。

【0006】

しかしながら、上記した従来のFPGAを用いた演算システムでは、ハードウェアとしてのモジュール分割は考えられていても、ソフトウェアとしてのモジュール分割は考えられていなかった。つまり、ソフトウェアとして1のプログラムモジュールから他のプログラムモジュールを呼び出し、呼び出したプログラムモジュールの実行を終了した後、元のプログラムモジュールに復帰するというように、複数のプログラムモジュールを適時実行していくことにより大規模プログラムの実行を可能とする仕組みは考えられていなかった。

【0007】

また、ソフトウェアとしてのモジュール分割を行うとしても、プログラムモジュールを呼び出すごとにFPGAのコンフィギュレーションを逐次書き換えるようにした場合、書き換えの時間が長いため、オーバーヘッドの多い非効率的な演算システムとなってしまう。

【0008】

このため、従来のFPGAを用いた演算システムで実行可能なプログラムは、実質的に1のみのモジュールで作成されたプログラムでなくてはならないという制約があった。つまり、大規模プログラムの実行が事実上不可能で、その適用範囲は限られるという問題があった。

【0009】

本発明は、上記した従来技術の問題点を解消するためになされたものであり、汎用のCPUを用いることなく、複数のプログラムモジュールからなる大規模プログラムの効率的な実行をハードウェアで直接的に実現した演算システムを提供することを目的とする。

【0010】

【課題を解決するための手段】

上記目的を達成するため、本発明の演算システムは、
ルックアップテーブルを含んだ複数のプログラムモジュールを各々記憶する複数のプログラム記憶手段と、
各前記プログラム記憶手段のいずれかを選択する選択手段と、
複数の論理回路を含み、前記選択手段に選択されたプログラム記憶手段に記憶されているプログラムモジュール中の命令に従った信号を前記複数の論理回路の1以上に入力することで、該プログラムモジュールに応じた演算を実行する論理演算手段と、
を備えることを特徴とする。

【0011】

上記演算システムでは論理演算手段のコンフィギュレーションの書き換えによるオーバーヘッドがほとんど生じないため、演算が効率的になり、従って高速になる。

【0012】

なお、前記選択手段は、たとえば、シフトレジスタより構成されていればよい。
また、各前記プログラム記憶手段が、プログラムモジュールを各々書き換え可能に記憶するものである場合、前記選択手段は、所定のプログラム記憶手段が記憶するプログラムモジュールを選択するものであって、各前記プログラムモジュールを、各前記プログラム記憶手段相互間でサイクリックに転送させるものであってもよい。

【0013】

各前記プログラム記憶手段は、プログラムモジュールを各々書き換え可能に記

憶するものであってもよい。

この場合、前記演算システムは、前記プログラム記憶手段にプログラムモジュールをロードするロード手段を更に備えるものとするれば、プログラムモジュールのデータ量の総量が、プログラム記憶手段の記憶容量より大きくても、このプログラムモジュールをすべて利用することが可能になる。

【0014】

前記演算システムは、

前記論理演算手段の内部状態を退避する退避手段と、

所定の条件が成立した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、他のプログラムモジュールを前記ロード手段にロードさせ、該他のプログラムモジュールを記憶したプログラム記憶手段を前記選択手段に選択させ、該他のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、元のプログラムモジュールに応じた演算の実行に復帰させる制御手段と、

を更に備えていてもよい。

このような構成を有する演算システムは、メモリにロードするプログラムモジュールを切り替えるときに、論理演算手段の内部状態を退避し、また、復元する仕組みを備えている。このため、複数のプログラムモジュールからなる大規模なプログラムも、メモリにロードするプログラムを切り替え、論理演算手段中の論理回路に入力する信号を変えていくことで、ハードウェア的に高速に演算を実行していくことができる。

【0015】

上記演算システムにおいて、

前記複数のプログラムモジュールのうちの少なくとも一部のプログラムモジュールは、他のプログラムモジュールを呼び出す機能を含むものであってもよい。

この場合において、上記演算システムは、

前記論理演算手段で演算を実行しているプログラムモジュール中の命令における他のプログラムモジュールの呼び出しを検出する呼び出し検出手段をさらに備えるものとすることができ、

前記制御手段は、前記呼び出し検出手段が他のプログラムモジュールの呼び出しを検出した場合に、前記論理演算手段の内部状態を前記退避手段に退避すると共に、呼び出し先のプログラムモジュールが前記プログラム記憶手段に記憶されていないときは該呼び出し先のプログラムモジュールを前記ロード手段にロードさせ、該呼び出し先のプログラムモジュールを記憶しているプログラム記憶手段を前記選択手段に選択させ、該呼び出し先のプログラムモジュールに応じた演算の実行を終了した後に前記退避手段に退避した内部状態を前記論理演算手段に戻してから、呼び出し元のプログラムモジュールに応じた演算の実行に復帰させるものとすることができる。

【0016】

ここで、前記制御手段によって実行が切り替えられるプログラムモジュール間において引数を受け渡すための引数受け渡し手段をさらに備えていてもよい。

【0017】

これらの仕組みをさらに備えることによって、モジュールの呼び出しを含む大規模なプログラムをハードウェア的に高速に実行することが可能となる。

【0018】

上記演算システムにおいて、

前記退避手段は、先入れ後出し方式のスタックによって構成されたものとすることができる。

【0019】

このようなスタックで構成される退避手段により、例えば、別のプログラムモジュールから呼び出されたプログラムモジュールが、さらに他のプログラムモジュールを呼び出すといった処理も可能となる。また、あるプログラムモジュールが、そのプログラムモジュール自身を呼び出す再帰型のプログラムを実行することも可能となる。

【0020】

上記演算システムにおいて、

前記プログラム記憶手段に記憶された各プログラムモジュールは、具体的には、たとえば、前記論理演算手段を構成する論理回路に入力する信号に応じたコー

ドによって構成されている。

【0021】

なお、各プログラムモジュール中の命令を構成するコードは、ハードウェア記述が可能な言語で記述されたソースプログラムをコンパイルすることによって得ることができる。この場合、モジュール別にソースプログラムを開発したり、モジュールの部品としての利用が可能となり、プログラムの開発期間を短縮することが可能となる。

【0022】

【発明の実施の形態】

以下、添付図面を参照して、本発明の実施の形態について説明する。

【0023】

図1は、この実施の形態にかかる演算システムの構成を示すブロック図である。図示するように、この演算システム1は、FPGAデータ記憶部2と、ローダ3と、FPGAデバイス4とから構成されている。FPGAデータ記憶部2には、複数のモジュールに分かれたFPGAデータモジュール2-1～2-nを記憶している。

【0024】

FPGAデータモジュール2-1～2-nは、それぞれハードウェア記述が可能なプログラム言語で記述されている複数のモジュールに分かれたソースプログラム5-1～5-nを、FPGAデバイス4の論理記述を行うべくコンパイラ6がコンパイルしたモジュール毎のデータである。FPGAデータモジュール2-1～2-nはそれぞれ、入力データの値と出力データの値との対応関係を示すルックアップテーブルの形式をとるデータを含んでいる。

ソースプログラム5-1～5-nのうちの少なくとも1のモジュールは、他のモジュールのソースプログラム5-1～5-nを呼び出す機能を含んでおり、FPGAデータモジュール2-1～2-nには、他のモジュールの呼び出しのためのデータも含まれている。

【0025】

ローダ3は、論理回路等より構成されており、FPGAデータ記憶部2に記憶

されたFPGAデータモジュール2-1~2-nをモジュール単位でFPGAデバイス4にロードする。

ローダ3によるFPGAデータモジュール2-1~2-nのロードの指示は、演算の実行の開始時に外部から与えられる他、FPGAデバイス4による演算の実行によっても与えられる。ローダ3は、ロードの指示1回につき、この指示に従って、後述のFPGAデータメモリ41a~41dのいずれかに、FPGAデータモジュールをロードする。

【0026】

FPGAデバイス4は、ローダ3によってロードされたFPGAデータモジュール2-1~2-nに従って論理構成を行い、外部からの入力データに所定の演算を施して出力データとして出力するもので、シフトレジスタ40と、ゲートアレイ43と、呼び出し検出部44と、退避スタック45と、引数受け渡し部46と、制御部47とを備えている。呼び出し検出部44、退避スタック45、引数受け渡し部46及び制御部47は、論理回路等より構成されている。

【0027】

シフトレジスタ40は、4個のFPGAデータメモリ41a~41dを含んでいる。FPGAデータメモリ41a~41dはそれぞれ半導体メモリあるいはフリップフロップ回路などより構成されており、ローダ3がロードしたFPGAデータモジュールを記憶するための記憶領域を有している。

【0028】

FPGAデータメモリ41a~41dは、たとえば、それぞれデータバスと、アドレスバスと、ライトイネーブル端子と、リードイネーブル端子とを備える。そして、各自のライトイネーブル端子にアクティブレベルの信号が供給されているとき、データバスに供給されたデータを、アドレスバスに供給されたアドレスが示す記憶領域に格納する。

なお、ローダ3は、ロードの指示を供給されるたびに、例えば、FPGAデータモジュールをロードする記憶領域を示すアドレスをFPGAデータメモリ41a~41dのアドレスバスに供給し、また、当該ロードの指示が示す特定のFPGAデータメモリのライトイネーブル端子にアクティブレベルの信号を供給して

、この状態で、FPGAデータメモリ41a～41dに、ロードすべきFPGAデータモジュールを供給するものとする。

【0029】

また、シフトレジスタ40は、FPGAデータメモリ41a～41dのうちデータを出力する対象であるFPGAデータメモリ1個を、制御部47より与えられる指示に従って、サイクリックに指定する（すなわち、指定する対象を循環シフトさせる）。そして、指定したFPGAデータメモリが格納しているデータを、データバスを介して出力する。

FPGAデータメモリ41a～41dが、上述したデータバス、アドレスバス及びリードイネーブル端子を備え、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出してデータバスを介して出力する場合、シフトレジスタ40は、FPGAデータメモリ41a～41dのうちいずれかのリードイネーブル端子にアクティブレベルの信号を供給するものとする。そして、指示を供給されるたびに、例えば、アクティブレベルの信号の供給先であるイネーブル端子を、計4個のリードイネーブル端子のうちからサイクリックに切り替えるものとする。なお、シフトレジスタ40は、複数のFPGAデータメモリのリードイネーブル端子に同時にアクティブレベルの信号を供給して、当該複数のFPGAデータメモリから一括してデータを読み出すことが可能なように構成されていてもよい。

【0030】

ゲートアレイ43は、AND、OR、NOTなどの複数のゲート回路43aと、演算の途中結果を内部状態として保持している複数のフリップフロップ43bとを含んでいる。各ゲート回路43aの出力論理は、FPGAデータメモリ41a～41dのうちシフトレジスタ40により指定されたものに記憶されたFPGAデータモジュールに従って変更される。また、各フリップフロップ43bは、所望のデータを外部から書き込むことができるようになっている。

【0031】

呼び出し検出部44は、FPGAデータメモリ41a～41dのうちセクタ

42により選択されたものに記憶されたFPGAデータモジュールに含まれる他のモジュールの呼び出しのためのデータを検出する。退避スタック45は、呼び出し検出部44によって他のモジュールの呼び出しのためのデータが検出されたとき、ゲートアレイ43中のフリップフロップ43bに保持されているデータと、呼び出し元のFPGAデータモジュールの識別データとを退避するためのスタックである。退避の手法としては、例えば、先入れ後出し方式を用いる。

【0032】

引数受け渡し部46は、モジュールの呼び出し、復帰の際において呼び出し元と呼び出し先のFPGAデータモジュール間における引数の受け渡しを行うものである。より詳細に説明すると、呼び出しの際には、呼び出し元のFPGAデータモジュールに従った演算の途中結果としてフリップフロップ43bの所定のものに保持されていたデータを、呼び出し先のFPGAデータモジュールに従った演算の入力（引数）として与える。復帰の際には、呼び出し先のFPGAデータモジュールに従った演算結果（戻り値）の出力データを、ゲートアレイ43中のフリップフロップ43bの所定のものに書き込む。

【0033】

制御部47は、呼び出し検出部44が他のモジュールの呼び出しのためのデータを検出した場合、当該呼び出しのためのデータの前までのFPGAデータモジュールに従った演算の途中結果としてフリップフロップ43bのそれぞれに保持されているデータと、呼び出し元のデータモジュールの識別データとを退避スタック45に退避させると共に、呼び出し先のFPGAデータモジュールに従った演算で使用するデータを保持するフリップフロップ43bのデータを、引数受け渡し部46に一時保持させる。

その後、FPGAデータメモリ41a～41dのうち呼び出し先のFPGAデータモジュールを記憶するものがあれば、シフトレジスタ40に、該当するFPGAデータメモリが指定されるまで指定する対象を繰り返し循環シフトさせ、なければ、ローダ3にロードさせる。そして、引数受け渡し部46に一時保持したデータをゲートアレイ43に入力データとして与える。

【0034】

制御部 4 7 は、また、呼び出された F P G A データモジュールに従った演算が終了したときに、その出力データを引数受け渡し部 4 6 に一時保持させる。

その後、退避スタック 4 5 に退避された呼び出し元のデータモジュールの識別データに従って、シフトレジスタ 4 0 に循環シフトを行わせることにより呼び出し元の F P G A データモジュールを指定させ、退避スタック 4 5 に退避されたデータをフリップフロップ 4 3 b に復帰させると共に、引数受け渡し部 4 6 に一時保持させたデータをフリップフロップ 4 3 b の所定のもの書き込ませる。

【 0 0 3 5 】

なお、F P G A デバイス 4 に外部から入力される入力データは、キーボードなどの入力装置から入力されるデータその他、磁気ディスク装置などの外部記憶装置から読み出されたデータであってもよい。また、F P G A デバイス 4 から外部に出力される出力データは、ディスプレイ装置などの出力装置から出力する他、外部記憶装置に書き込むものであってもよく、さらに、周辺機器を制御するための制御データであってもよい。

【 0 0 3 6 】

以下、この実施の形態にかかる演算システムにおける動作について、具体的な例に基づいて説明する。ここでは、F P G A データモジュール 2 - 1 が最初にロードされるものとし、F P G A データモジュール 2 - 1 は、F P G A データモジュール 2 - n を呼び出すものとする。また、退避スタック 4 5 は、先入れ後出し方式でデータの退避を行うものとする。

【 0 0 3 7 】

F P G A データモジュール 2 - 1 が F P G A データメモリ 4 1 a ~ 4 1 d のいずれかにロードされ（以下では、理解を容易にするため、F P G A データメモリ 4 1 a にロードされたものとする）、セレクト 4 2 が F P G A データメモリ 4 1 a を選択すると、これに従ったレベルの信号がゲート回路 4 3 a に入力され、ゲートアレイ 4 3 を構成するゲート回路 4 3 a が論理構成される。そして、ゲートアレイ 4 3 に外部からの入力データが入力されることによって、F P G A データモジュール 2 - 1 に応じた演算がゲートアレイ 4 3 において実行される。

【 0 0 3 8 】

一方、呼び出し検出部 4 4 は、FPGA データメモリ 4 1 a にロードされた FPGA データモジュール 2 - 1 に FPGA データモジュール 2 - n を呼び出すためのデータが含まれていることを検出し、その旨を制御部 4 7 に通知する。制御部 4 7 は、その呼び出しにかかる部分の直前までの演算の途中結果としてフリップフロップ 4 3 b に保持されているデータ（ゲートアレイ 4 3 の内部状態）を、呼び出し元の FPGA データモジュール 2 - 1 を識別するためのデータと共に退避スタック 4 5 の一番上に退避させる。また、フリップフロップ 4 3 b に保持されているデータのうちで呼び出し先の FPGA データモジュール 2 - n に引数として渡すものを、引数受け渡し部 4 6 に一時保存させる。

【0039】

その後、制御部 4 7 は、FPGA データモジュール 2 - n が FPGA データメモリ 4 1 b ~ 4 1 d のいずれかにロードされているか否かを判別する。（具体的には、たとえば、FPGA データモジュール 2 - n をロードする旨の指示を既に自らロード 3 に供給したか否かを判別する。）そして、ロードされていなければ、ロード 3 を制御し、呼び出し先である FPGA データモジュール 2 - n を FPGA データメモリ 4 1 b ~ 4 1 d のいずれかにロードさせ（以下では、理解を容易にするため、FPGA データメモリ 4 1 b にロードされたものとする）、シフトレジスタ 4 0 に FPGA データメモリ 4 1 b を指定させる。一方、ロードされていれば、シフトレジスタ 4 0 に、FPGA データモジュール 2 - n がロードされている FPGA データメモリ（以下では、理解を容易にするため、FPGA データメモリ 4 1 b がこれに該当するものとする）を指定させる。

【0040】

FPGA データモジュール 2 - n が FPGA データメモリ 4 1 b にロードされ、シフトレジスタ 4 0 が FPGA データメモリ 4 1 b を指定すると、これに従ったレベルの信号がゲート回路 4 3 a に入力され、ゲートアレイ 4 3 を構成するゲート回路 4 3 a が論理構成される。また、引数受け渡し部 4 6 に引数として一時保存されたデータが、入力データとしてゲートアレイ 4 3 に入力され、FPGA データモジュール 2 - n に応じた演算がゲートアレイ 4 3 において実行される。

【0041】

この演算が終了すると、制御部 4 7 は、ゲートアレイ 4 3 からの出力データを呼び出し元の F P G A データモジュール 2 - 1 に渡す引数として引数受け渡し部 4 6 に一時保存させる。制御部 4 7 は、さらに退避スタック 4 5 の一番上に退避されたデータを参照することでシフトレジスタ 4 0 を制御して循環シフトをおこなわせ、呼び出し元の F P G A データモジュール 2 - 1 を記憶する F P G A データメモリ 4 1 a を再び指定させる。

【 0 0 4 2 】

呼び出し元の F P G A データモジュール 2 - 1 がロードされた F P G A データメモリ 4 1 a が再び指定されると、制御部 4 7 は、退避スタック 4 5 の一番上に退避されていた内部状態のデータをフリップフロップ 4 3 b のそれぞれに書き戻し、ゲートアレイ 4 3 の内部状態を復元させる。さらに、引数受け渡し部 4 6 に引数として一時保存されていたデータをフリップフロップ 4 3 b の所定のものに書き込む。この状態でゲートアレイ 4 3 において F P G A データモジュール 2 - 1 に従った演算が再開され、最終的な演算結果が出力データとして出力されることとなる。

【 0 0 4 3 】

なお、F P G A データモジュール 2 - 1 から呼び出された F P G A データモジュール 2 - n が、さらに他の F P G A データモジュールを呼び出すものであっても演算を実行することができる。F P G A データモジュール 2 - n がさらに他のモジュールを呼び出すことを呼び出し検出部 4 4 が検出した場合にも、制御部 4 7 は、上記と同じような制御を行うものとすればよい。

【 0 0 4 4 】

以上説明したように、この実施の形態にかかる演算システムでは、ゲートアレイ 4 3 の内部状態（フリップフロップ 4 3 b が保持するデータ）を退避スタック 4 5 に退避した後に、実行中のモジュールとは異なる F P G A データモジュールが F P G A データメモリ 4 1 a ~ 4 1 d のいずれかにロードされ、ロードされた F P G A データメモリの記憶内容が出力されるようにしている。また、退避スタック 4 5 に退避した状態をゲートアレイ 4 3 に復元してから元のモジュールに復帰することができるようになっている。

このため、各FPGAデータモジュールをFPGAデータメモリに適宜ロードしていくことによって、複数のモジュールからなる大規模なプログラムを、各モジュールに対応してゲート回路43a間の論理構成を変化させてハードウェア的に実行することができ、従来のCPUを用いた演算システムに比べて高速に演算を実行することができる。

【0045】

さらに、この演算システムはFPGAデータメモリを複数備えており、制御に従って適宜選択される。このため、コンフィギュレーションの書き換えによるオーバーヘッドが生じず、FPGAデータメモリが単一でFPGAデータモジュールの呼び出しがある毎に新たなFPGAデータモジュールがロードし直される構成に比べても演算が効率的で高速になる。

【0046】

また、FPGAデータモジュール2-1～2-nのうちの少なくとも1のモジュールが他のモジュールを呼び出すためのデータを含んでいるが、このような他のモジュールの呼び出しを含むFPGAデータモジュールがFPGAデータメモリにロードされた場合に、これを呼び出し検出部44が検出している。そして、この検出結果に基づいて、退避スタック45へのゲートアレイ43の内部状態（フリップフロップ43bが保持するデータ）の退避、引数受け渡し部46を介した引数の受け渡しを行っている。また、呼び出し先のモジュールに従った演算が終了したときに、退避スタック45に退避した内部状態の復元、引数受け渡し部46を介した呼び出し元のモジュールへの引数の受け渡しを行っている。このような仕組みを備えることによって、モジュールの呼び出しを含む大規模なプログラムをハードウェア的に実行することが可能となる。

【0047】

また、呼び出し検出部44が他のモジュールの呼び出しを検出したときに、ゲートアレイ43の内部状態（フリップフロップ43bが保持するデータ）を退避するのは、先入れ後出し方式の退避スタックである。このため、他のモジュールから呼び出されたモジュールがさらに他のモジュールを呼び出すようなプログラムを実行することもできる。さらに、実行中のモジュールが自身を呼び出す再帰

型のプログラムを実行することもできる。

【0048】

さらに、FPGAデータモジュール2-1～2-nは、モジュール分割されたソースプログラム5-1～5-nをそれぞれコンパイラ6によってコンパイルしたものである。以上のような特徴を有することによって、この演算システムにおいて実行すべきプログラムは、モジュール別にソースプログラムの開発を進めたり、ソースプログラムの各モジュールを部品として利用したりすることが可能となり、その開発期間を短縮することができる。

【0049】

本発明は、上記の実施の形態に限られず、種々の変形、応用が可能である。以下、本発明に適用可能な上記の実施の形態の変形態様について説明する。

【0050】

上記の実施の形態では、ローダ3は、FPGAデータ記憶部2に記憶されたいずれかのFPGAデータモジュール2-1～2-nを、そのままFPGAデータメモリ41a～41dにロードするものとしていた。これに対して、FPGAデータモジュール2-1～2-nがマクロを含み、FPGAデータ記憶部2にマクロデータを記憶させておき、ローダ3がFPGAデータメモリ41a～41dにロードする際に、マクロ展開をするものとしてもよい。

【0051】

また、FPGAデータメモリ41a～41dはシフトレジスタ40とは別個のものとして、それぞれRAM (Random Access Memory) 等のランダムアクセス可能なメモリより構成されていてもよい。

そしてこの場合、例えば図2に示すように、この演算システムは、シフトレジスタ40に代えてセクタ42を備えるものとしてもよい。

【0052】

この場合、FPGAデータメモリ41a～41dは、たとえば、それぞれデータ／アドレスバスと、ライトイネーブル端子と、リードイネーブル端子とを備えるものとする。そして、各自のライトイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスにまず供給されたアドレスが示す

記憶領域に、このデータ／アドレスバスに次に供給されたデータを格納する。また、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出し、このデータ／アドレスバスを介して出力するものとする。

FPGAデータメモリ41a～41dがこのような構成を有している場合、ロード3は、ロードの指示を供給されるたびに、FPGAデータメモリ41a～41dのうちこの指示が示すもののライトイネーブル端子にアクティブレベルの信号を供給し、この状態で、FPGAデータメモリ41a～41dに、ロードすべきFPGAデータモジュールを供給すればよい。

【0053】

セクタ42は、シフトレジスタ等より構成されており、FPGAデバイス4による演算の実行によって制御部47より与えられる指示に従い、FPGAデータメモリ41a～41dのうちこの指示が示すもの1個を指定する。（すなわち、ゲートアレイ43や呼び出し検出部44がデータを参照することが可能な状態とする。）

FPGAデータメモリ41a～41dが、上述したデータ／アドレスバス及びリードイネーブル端子を備え、各自のリードイネーブル端子にアクティブレベルの信号が供給されているとき、データ／アドレスバスに供給されたアドレスが示す記憶領域に格納されているデータを読み出してこのデータ／アドレスバスを介して出力する場合、セクタ42は、指示を供給されるたびに、FPGAデータメモリ41a～41dのうちこの指示が示すもののリードイネーブル端子にアクティブレベルの信号を供給すればよい。

【0054】

また、FPGAデータメモリ41a～41dがデータを書き換え可能に記憶するメモリより構成されている場合、FPGAデータメモリ41a～41dは、データの入力及び出力に共用されるデータバスに代えて、書き込む対象のデータを入力するための入力ポートと、読み出したデータを出力するための出力ポートとを、互いに別個のものとして備えるようにしてもよい。

そしてこの場合、FPGAデータメモリ41a～41dは、ひとつのFPGA

データメモリの出力ポートが他のFPGAデータメモリの入力ポートに接続されるようにして、全体として環をなすように接続されていてもよい。

具体的には、例えば図3に示すように、FPGAデータメモリ41aの出力ポートOUTがFPGAデータメモリ41bの入力ポートINに接続され、FPGAデータメモリ41bの出力ポートOUTがFPGAデータメモリ41cの入力ポートINに接続され、FPGAデータメモリ41cの出力ポートOUTがFPGAデータメモリ41dの入力ポートINに接続され、FPGAデータメモリ41dの出力ポートOUTがFPGAデータメモリ41aの入力ポートINに接続されていてもよい。

【0055】

そして、シフトレジスタ40は、データを出力させる対象のFPGAデータメモリをサイクリックに指定する代わりに、FPGAデータメモリ41a～41dに記憶されているデータ自体をFPGAデータメモリ41a～41d相互間でサイクリックに転送させ、ゲートアレイ43には、所定のFPGAデータメモリが記憶しているデータを供給するものとしてもよい。

具体的には、例えば図3に模式的に示すように、シフトレジスタ40がデータバスを介して出力するデータは常にFPGAデータメモリ41aが記憶するデータであるものとし、シフトレジスタ40は、制御部47から指示が1回与えられる毎に、FPGAデータメモリ41aが記憶していた内容をFPGAデータメモリ41bに記憶させ、FPGAデータメモリ41bが記憶していた内容をFPGAデータメモリ41cに記憶させ、FPGAデータメモリ41cが記憶していた内容をFPGAデータメモリ41dに記憶させ、FPGAデータメモリ41dが記憶していた内容をFPGAデータメモリ41aに記憶させる、という動作をFPGAデータメモリ41a～41dに行わせるようにしてもよい。

【0056】

また、FPGAデータモジュール2-1～2-nは、ロード3を介さず予めFPGAデータメモリに格納されることとして、FPGAデータメモリが形成するルックアップテーブル自体をシフトレジスタが循環シフトさせるようにしてもよい。

この場合、図4に示すように、この演算システムはローダ3を省略してもよく、従って制御部47はローダ3を制御する必要はない。一方、シフトレジスタ40は、n個のFPGAデータモジュールを格納するためのn個のFPGAデータメモリ41-1～41-nを備え、FPGAデータメモリ41-1～41-nを指定の対象として循環シフトさせるものとする。

FPGAデータメモリ41-1～41-nの構成は、上述のFPGAデータメモリ41a～41dと実質的に同一であるものとする。ただし、FPGAデータメモリ41-1～41-nは、必ずしも記憶内容を書き換え可能である必要はない。

なお、図4の構成においても、FPGAデータメモリ41-1～41-nはシフトレジスタ40とは別個のものとしてランダムアクセス可能なメモリより構成されていてもよい。この場合、この演算システムは、例えば、シフトレジスタ40に代えて、図2に示すものと実質的に同一のセクタ42を備えていればよい。

【0057】

また、図1～図4の構成では、ソースプログラム5-1～5-nをそれぞれコンパイルしたFPGAデータモジュール2-1～2-nを、FPGAデバイス4のFPGAデータメモリ41a～41dに適宜ロードしていくものとしていた。これに対して、ソースプログラム5-1～5-nをそのままロードするようにした演算システムを構成することもできる。図5は、このような場合の演算システムの構成を示す。

【0058】

この演算システムでは、ローダ3'は、制御部47'からの指示に基づいて、プログラム記憶部5に記憶されたモジュール別のソースプログラム5-1～5-nを適宜メモリ41a'～41d'にロードし、シフトレジスタ40'は、制御部47'からの指示に基づいて、メモリ41a'～41d'のうち、実行すべきソースプログラムを記憶しているものを指定する。インタプリタ48は、シフトレジスタ40'に指定されたメモリにロードされたソースプログラム中の命令を1命令ずつ順次解釈し、その解釈結果に従ってゲートアレイ43'を構成するゲ

ート回路 43 a に論理構成を行わせるべく所定の信号を出力する。解釈の結果、他のモジュールのソースプログラムを呼び出す命令であった場合には、その旨を制御部 47' に通知する。

【0059】

制御部 47' は、他のモジュールの呼び出しが通知されると、ゲートアレイ 43' の内部状態（フリップフロップ 43 b に保持されているデータ）と、呼び出し元のソースプログラムのモジュールを識別するためのデータと、次に実行をすべき命令を示すデータを退避スタック 45 に退避すると共に、フリップフロップ 43 b に保持されているデータのうち呼び出し先のモジュールに引数として渡すものを、引数受け渡し部 46 に一時保存させる。そして、呼び出し先のソースプログラムが既にロードされているか否かを判別し、ロードされていなければ、ロード 3' に呼び出し先のソースプログラムをロードさせ、シフトレジスタ 40' に、呼び出し先のソースプログラムがロードされたメモリを指定させて、引数受け渡し部 46 に一時保存されたデータを入力データとしてゲートアレイ 43' に与える。一方、ロードされていれば、呼び出し先のソースプログラムがロードされたメモリをシフトレジスタ 40' に指定させて、引数受け渡し部 46 に一時保存されたデータを入力データとしてゲートアレイ 43' に与える。

【0060】

また、呼び出し先のソースプログラムに従った演算が終了すると、ゲートアレイ 43' からの出力データを呼び出し元のモジュールに渡す引数として引数受け渡し部 46 に一時保存させる。そして、退避スタック 45 に退避されたデータに従って、呼び出し元のソースプログラムを記憶するメモリを再びシフトレジスタ 40' に指定させ、退避スタック 45 に退避された内部状態をフリップフロップ 43 b に戻し、引数受け渡し部 46 に一時保存された引数をフリップフロップ 43 b のうちの所定のものには書き込ませる。そして、退避スタック 45 に退避されたデータに基づいて呼び出し元のモジュールのソースプログラムに従った演算を再開させる。

【0061】

なお、インタプリタ 48 は、複数のゲート回路の組み合わせによるハードウェア

アで構成することができ、その出力によってゲートアレイ 4 3' に含まれるゲート回路 4 3 a の論理構成を、演算の実行速度にほとんど影響を与えることなく高速に行うことができる。また、ここでのゲートアレイ 4 3' は、ソースプログラム中の各命令を終了したときのデータをフリップフロップ 4 3 b の所定のものに保持させることで、各命令を順次実行していくことができる。

【 0 0 6 2 】

以上のようにインタプリタ 4 8 を含む構成とすることによって、ソースプログラム 5 - 1 ~ 5 - n をモジュール別に順次 F P G A デバイス 4' にロードしていくことが可能となる。このため、F P G A デバイス 4' の構成に合わせたコンパイラがなくても、複数のモジュールからなる大規模なプログラムに従った演算を、ハードウェア的に高速に行うことが可能となる。

さらに、図 5 の演算システムはメモリを複数備えており、制御に従って適宜選択される。このため、コンフィギュレーションの書き換えによるオーバーヘッドが生じず、メモリが単一で F P G A データモジュールの呼び出しがある毎に新たなソースプログラムがロードし直される構成に比べても演算が効率的で高速になる。

【 0 0 6 3 】

なお、図 5 の構成においても、メモリ 4 1 a' ~ 4 1 d' はシフトレジスタ 4 0' とは別個のものとして、ランダムアクセス可能なメモリより構成されていてもよい。この場合、例えば図 6 に示すように、この演算システムは、シフトレジスタ 4 0' に代えて、図 2 の構成におけるセクタ 4 2 と実質的に同一の動作を行うセクタ 4 2' を備えるものとしてもよい。

【 0 0 6 4 】

また、メモリ 4 1 a' ~ 4 1 d' は図 3 に示す F P G A データメモリ 4 1 a ~ 4 1 d と実質的に同一の構成を有していてもよく、この場合、シフトレジスタ 4 0' は、図 3 の構成におけるシフトレジスタ 4 0 と実質的に同一の動作を行えばよい。

【 0 0 6 5 】

また、ソースプログラム 5 - 1 ~ 5 - n は、ローダ 3' を介さず予めメモリに

格納することとしてもよい。

この場合、図7に示すように、この演算システムはローダ3'を省略できる。一方、シフトレジスタ40'は、n個のソースプログラムを格納するためのn個のメモリ41-1'～41-n'を備え、これらn個のメモリを指定の対象として循環シフトさせるものとする。メモリ41-1'～41-n'の構成は、上述のFPGAデータメモリ41-1～41-nと実質的に同一であるものとする。

【0066】

なお、図7の構成においても、メモリ41-1'～41-n'はシフトレジスタ40'とは別個のものとしてランダムアクセス可能なメモリより構成されていてもよい。この場合、この演算システムは、例えば、シフトレジスタ40'に代えて、図6に示すものと実質的に同一のセクタ42'を備えていればよい。

【0067】

【発明の効果】

以上説明したように本発明によれば、複数のプログラムモジュールからなる大規模なプログラムであっても、各プログラムモジュールを適時メモリにロードしていく仕組みを有するので、該プログラムに応じた演算の実行をハードウェアで実現することが可能となる。また、コンフィギュレーションの書き換えによるオーバーヘッドの影響がなく演算が効率的となる。

【図面の簡単な説明】

【図1】

本発明の実施の形態にかかる演算システムの構成を示すブロック図である。

【図2】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

。

【図3】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

。

【図4】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である。

。

【図 5】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である

。

【図 6】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である

。

【図 7】

本発明の他の実施の形態にかかる演算システムの構成を示すブロック図である

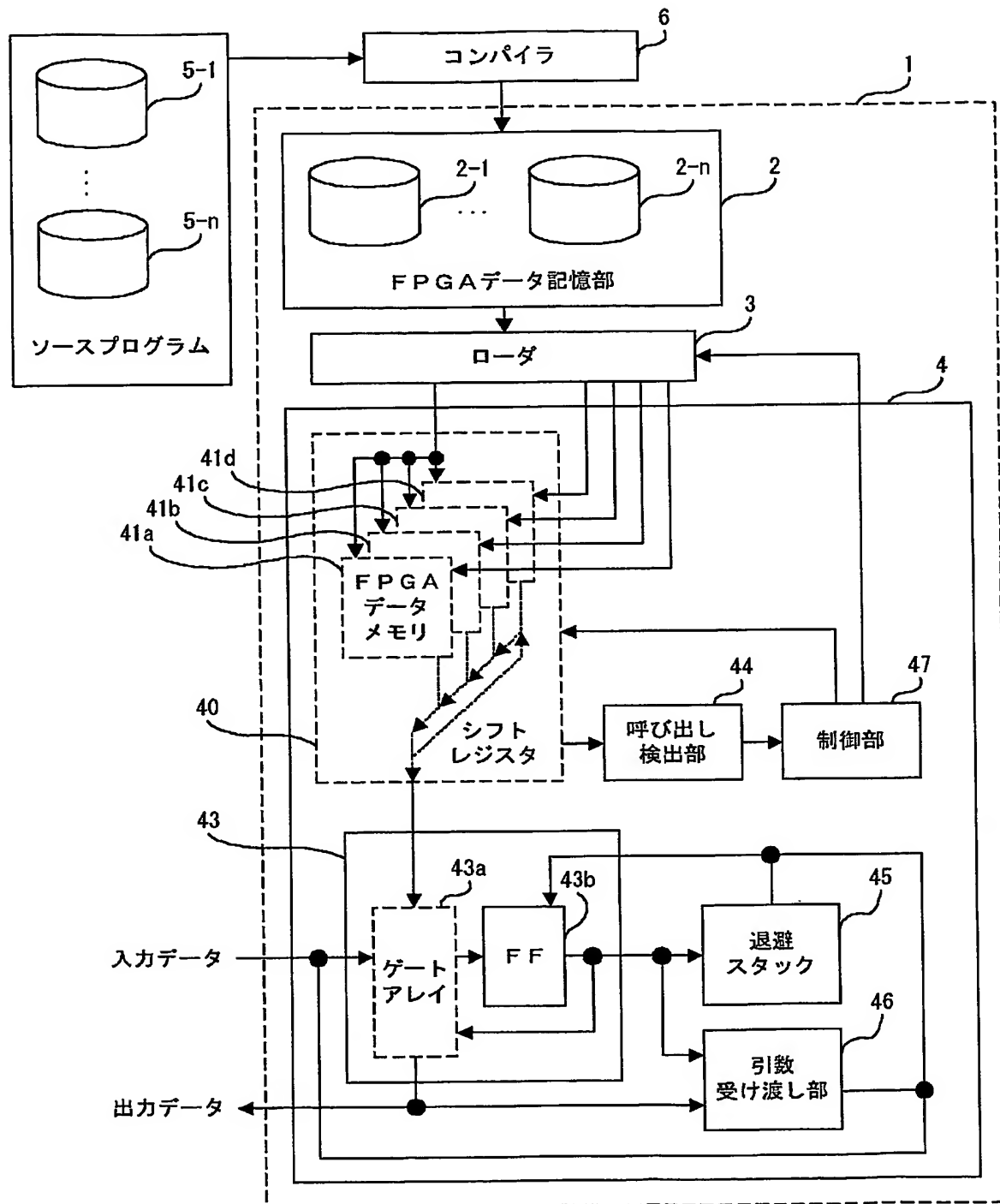
。

【符号の説明】

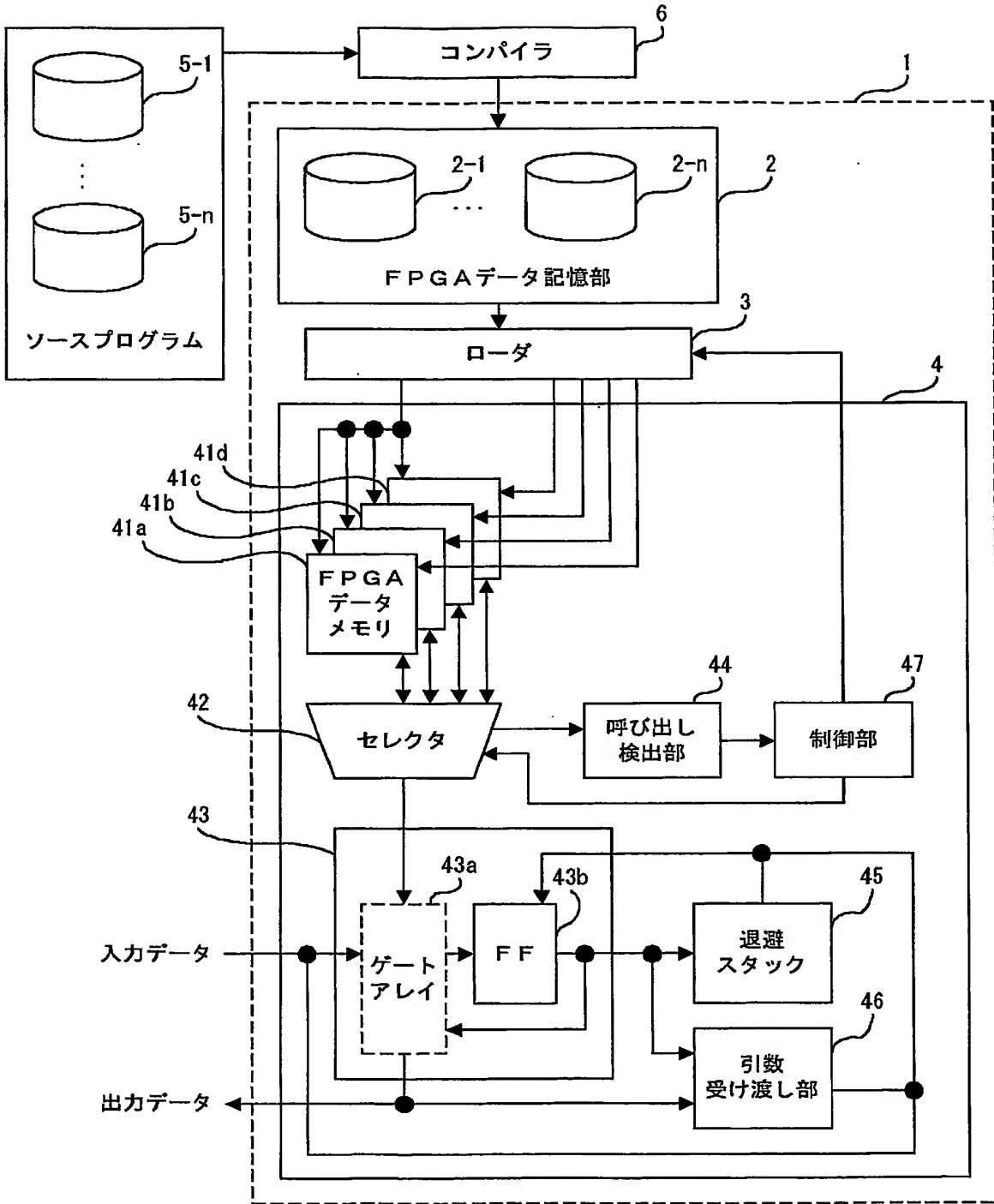
- 1 演算システム
- 2 F P G A データ記憶部
- 3 ローダ
- 4 F P G A デバイス
- 6 コンパイラ
- 2-1 ~ 2-n F P G A データモジュール
- 40 シフトレジスタ
- 41a ~ 41d、41-1 ~ 41-n F P G A データメモリ
- 42 セレクタ
- 43 ゲートアレイ
- 43a ゲート回路
- 43b フリップフロップ
- 44 呼び出し検出部
- 45 退避スタック
- 46 引数受け渡し部
- 47 制御部
- 48 インタプリタ
- 5-1 ~ 5-n ソースプログラム

【書類名】 図面

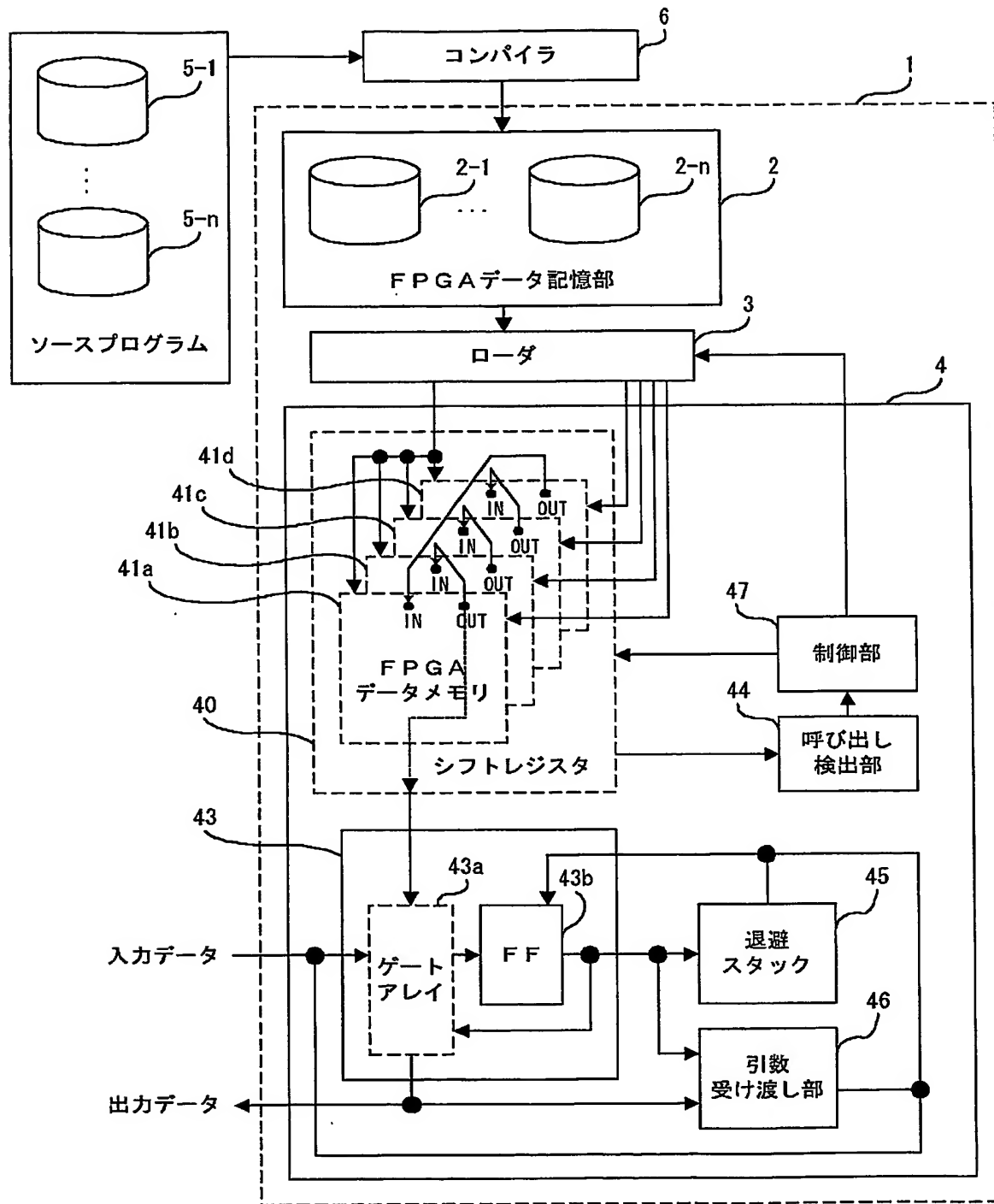
【図 1】



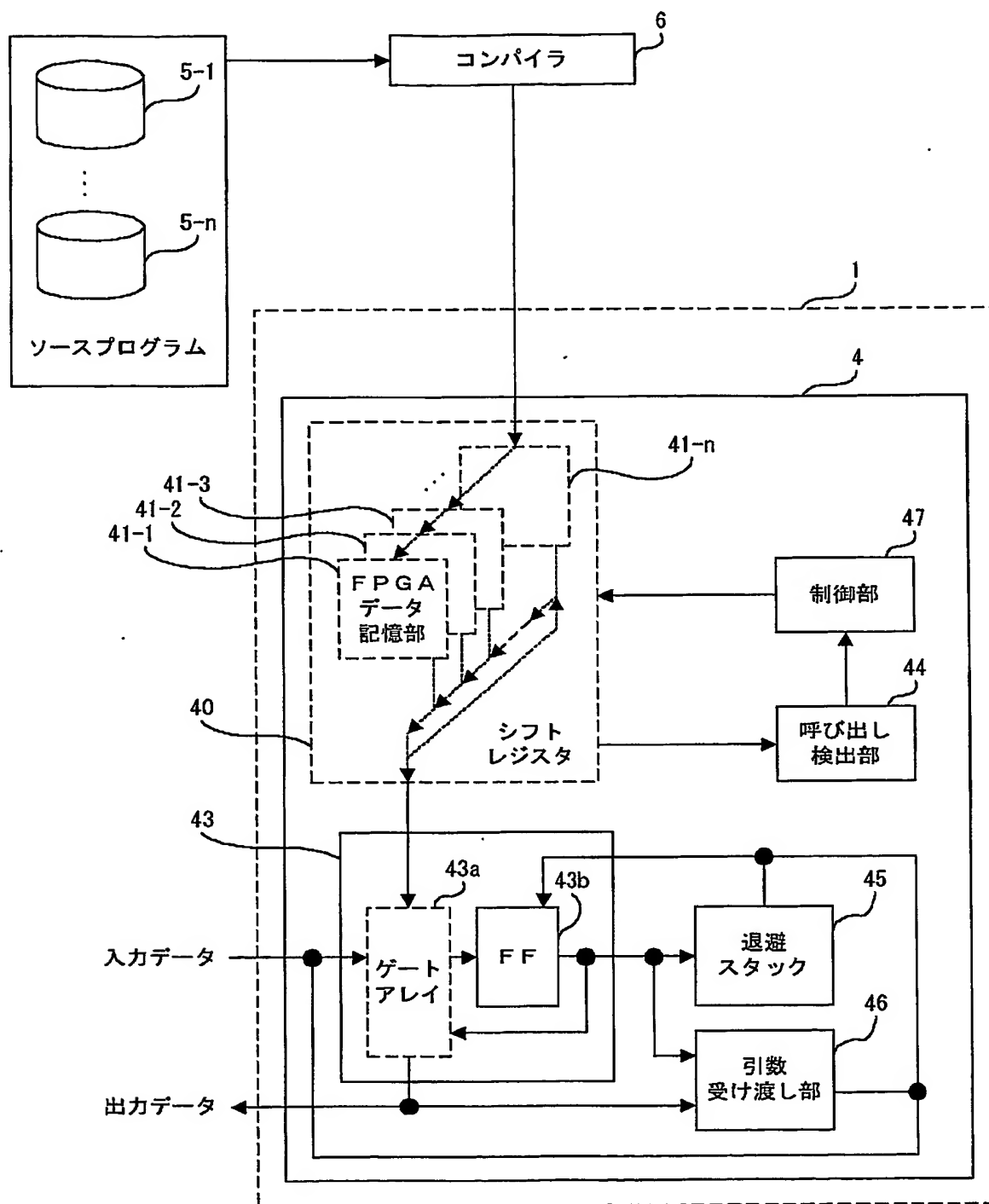
【図2】



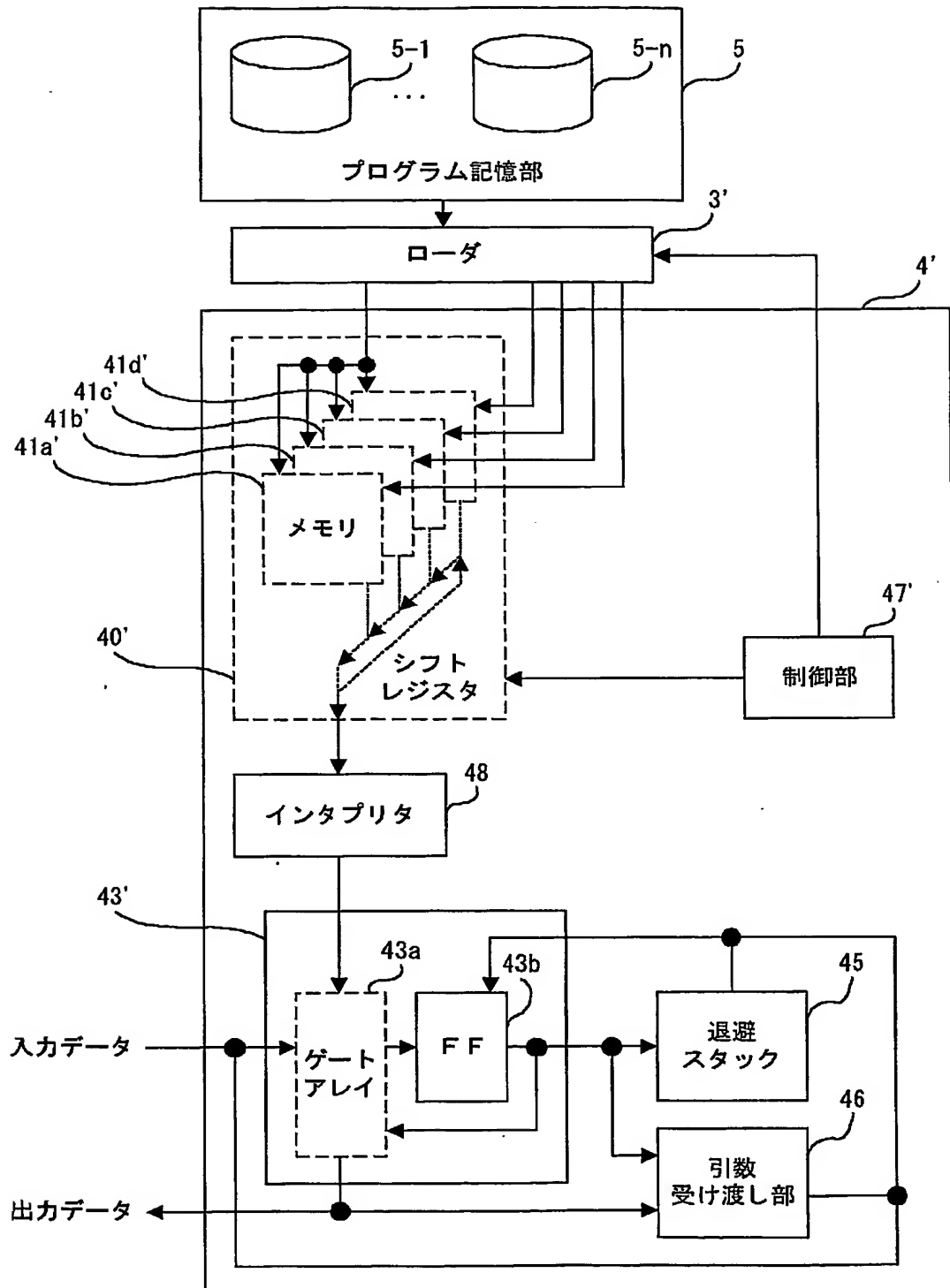
【図3】



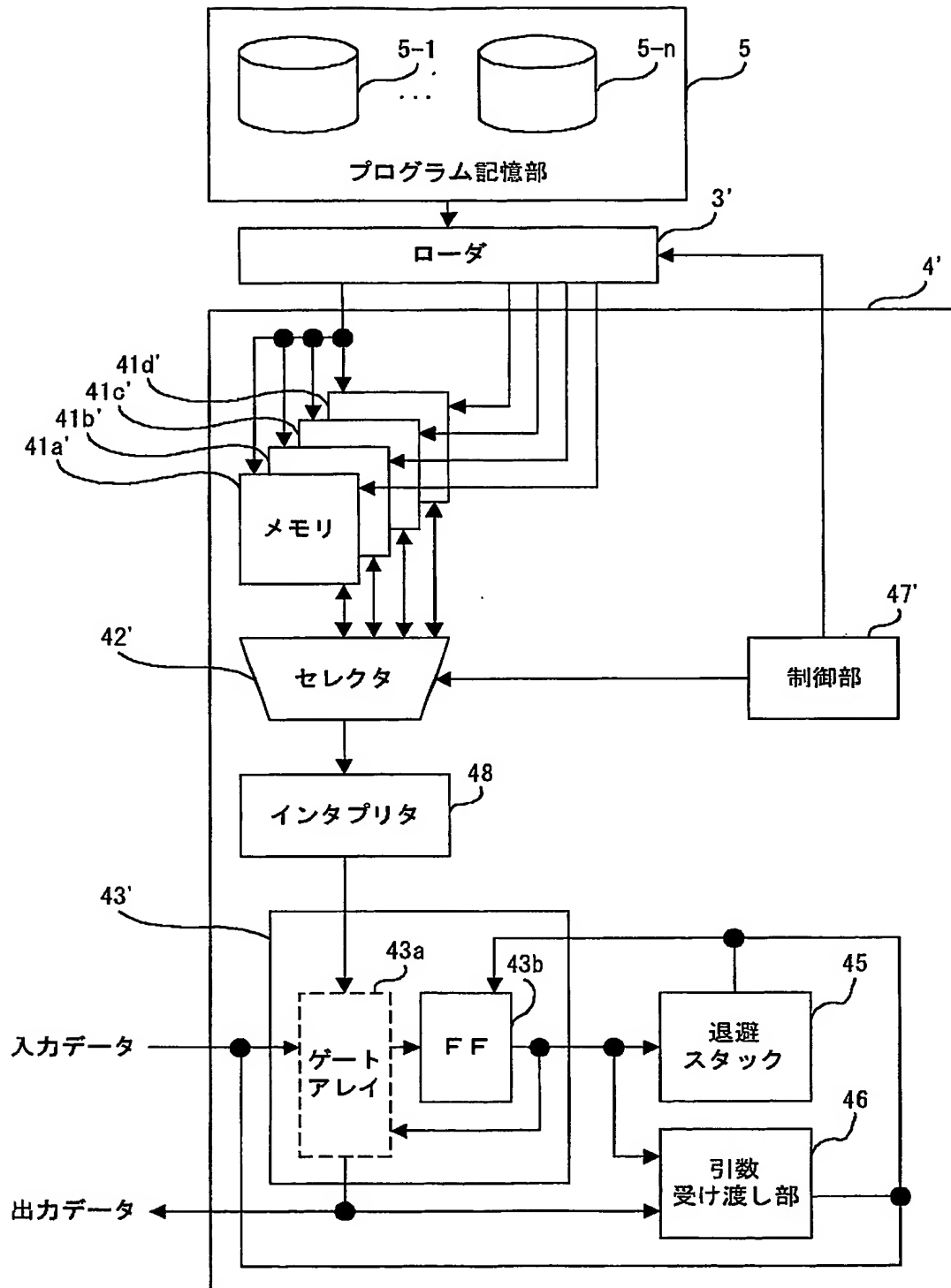
【図 4】



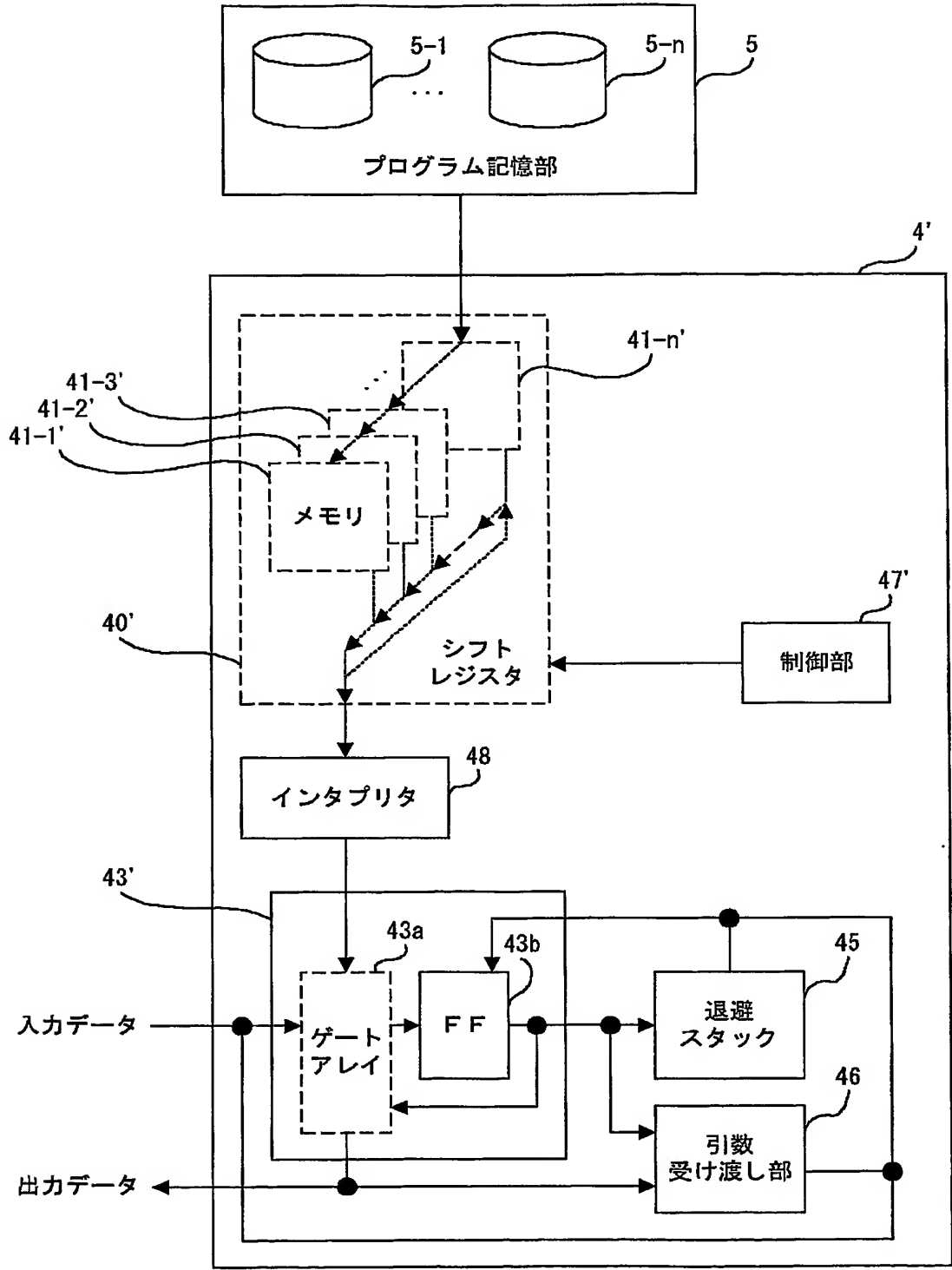
【図 5】



【図 6】



【図 7】



【書類名】 要約書

【要約】

【課題】 汎用のCPUを用いることなく、複数のプログラムモジュールからなる大規模プログラムの効率的な実行をハードウェアで直接的に実現する。

【解決手段】 ゲートアレイ43は、シフトレジスタ40に指定されたFPGAデータメモリに記憶されたFPGAデータモジュールに従ってハードウェア的に演算を行う。このFPGAデータメモリに記憶されたモジュールが他のモジュールを呼び出すものであることが呼び出し検出部44によって検出されると、フリップフロップ43bに保持された演算の途中結果のデータが退避スタック45に退避され、呼び出し先のモジュールに渡す引数が引数受け渡し部46に一時保存される。その後、呼び出し先のモジュールを記憶したFPGAデータメモリをシフトレジスタ40に指定させ、呼び出し元のモジュールに復帰する際には、退避スタック45に退避されたデータがフリップフロップ43bに書き戻される。

【選択図】 図1

【書類名】 手続補正書

【提出日】 平成14年 1月24日

【あて先】 特許庁長官 殿

【事件の表示】

【出願番号】 特願2001-401462

【補正をする者】

【識別番号】 500323188

【氏名又は名称】 東京エレクトロニクス株式会社

【補正をする者】

【識別番号】 501186977

【氏名又は名称】 西原 明法

【代理人】

【識別番号】 100095407

【弁理士】

【氏名又は名称】 木村 満

【手続補正 1】

【補正対象書類名】 特許願

【補正対象項目名】 発明者

【補正方法】 変更

【補正の内容】

【発明者】

【住所又は居所】 神奈川県横浜市都筑区東方町 1 番地 東京エレクトロニクス株式会社内

【氏名】 三田 高司

【発明者】

【住所又は居所】 神奈川県川崎市中原区今井仲町 3 7 4 - 4 - 5 0 5

【氏名】 西原 明法

【その他】 変更（追加）の理由は、代理人の不注意による記入漏れであります。

【プルーフの要否】 要

特願 2 0 0 1 - 4 0 1 4 6 2

出 願 人 履 歴 情 報

識別番号 [5 0 0 3 2 3 1 8 8]

1. 変更年月日 2 0 0 0 年 7 月 7 日

[変更理由] 新規登録

住 所 神奈川県横浜市都筑区東方町 1 番地
氏 名 東京エレクトロニクス株式会社

特願 2001-401462

出願人履歴情報

識別番号

[501186977]

1. 変更年月日

2001年 5月10日

[変更理由]

新規登録

住 所

神奈川県川崎市中原区今井仲町 374-4-505

氏 名

西原 明法

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.